AF

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission

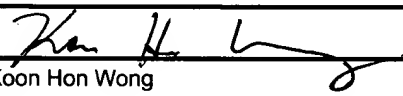
Application Number	09/637,078
Filing Date	August 11, 2000
First Named Inventor	Wang
Art Unit	2193
Examiner Name	W. Wood
Attorney Docket Number	YOR92000-0415US1 (8728-407)

ENCLOSURES *(Check all that apply)*

- [X] Fee Transmittal Form
  - [ ] Fee Attached
- [ ] Amendment/Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Reply to Missing Parts/ Incomplete Application
  - [ ] Reply to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
  - [ ] Landscape Table on CD
- [ ] After Allowance Communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [X] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [ ] Other Enclosure(s) (please Identify below):

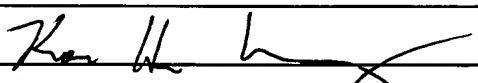
Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	F. Chau & Associates, LLC		
Signature			
Printed name	Koon Hon Wong		
Date	July 11, 2005	Reg. No.	48,459

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature			
Typed or printed name	Koon Hon Wong	Date	July 11, 2005

PTO/SB/17 (12-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL
For FY 2005

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 500.00

Complete if Known	
Application Number	09/637,078
Filing Date	August 11, 2000
First Named Inventor	Wang
Examiner Name	W. Wood
Art Unit	2193
Attorney Docket No.	YOR9-2000-0415US1 (8728-407)

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____

☒ Deposit Account Deposit Account Number: 50-0150 Deposit Account Name: IBM/YORKTOWN HEIGHTS

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☒ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, **except for the filing fee**

☒ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☐ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
Application Type	**Fee ($)**	**Small Entity Fee ($)**	**Fee ($)**	**Small Entity Fee ($)**	**Fee ($)**	**Small Entity Fee ($)**	**Fees Paid ($)**
Utility	300	150	500	250	200	100	
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description	**Fee ($)**	**Small Entity Fee ($)**
Each claim over 20 or, for Reissues, each claim over 20 and more than in the original patent	50	25
Each independent claim over 3 or, for Reissues, each independent claim more than in the original patent	200	100
Multiple dependent claims	360	180

Total Claims		**Extra Claims**	**Fee ($)**		**Fee Paid ($)**
_____	- 20 or HP =	_____	x _____	=	_____

HP = highest number of total claims paid for, if greater than 20

Indep. Claims		**Extra Claims**	**Fee ($)**		**Fee Paid ($)**
_____	- 3 or HP =	_____	x _____	=	_____

HP = highest number of independent claims paid for, if greater than 3

Multiple Dependent Claims	
Fee ($)	**Fee Paid ($)**
_____	_____

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets		**Extra Sheets**		**Number of each additional 50 or fraction thereof**	**Fee ($)**		**Fee Paid ($)**
_____	- 100 =	_____	/ 50 =	_____ (round **up** to a whole number)	x _____	=	_____

4. OTHER FEE(S)

	Fees Paid ($)
Non-English Specification, $130 fee (no small entity discount)	
Other: Appeal Brief	500.00

SUBMITTED BY

Signature		Registration No. (Attorney/Agent) 48,459	Telephone 516-692-8888
Name (Print/Type)	Koon Hon Wong		Date July 11, 2005

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appellants:	Wang et al.	Examiner: W. Wood
Serial No:	09/637,078	Group Art Unit: 2193
Filed:	August 11, 2000	Docket: YOR9-2000-0415US1 (8728-407)
For:	METHOD AND APPARATUS FOR PROFILING COMPUTER PROGRAM EXECUTION	

APPEAL BRIEF

This is an Appeal from the Final Office Action mailed December 23, 2004 (Paper No. 112904), finally rejecting claims 1, 3-16, 18-30 and 32-42. Applicants appeal pursuant to the Notice of Appeal received by the USPTO on May 10, 2005 and submit this appeal brief.

Appeal from Group 2193

F. Chau & Associates, LLC
130 Woodbury Road
Woodbury, NY 11797
TEL: (516) 692-8888
FAX: (516) 692-8889
Attorneys for Appellants

TABLE OF CONTENTS

1. Real Party in Interest

The real party in interest is INTERNATIONAL BUSINESS MACHINES CORPORATION, the assignee of the entire right, title and interest in and to the subject application by virtue of an assignment of record.

2. Related Appeals and Interferences

None.

3. Status of Claims

Claims 1, 3-16, 18-30 and 32-42 are pending, stand rejected and are under appeal.

A copy of the claims 1, 3-16, 18-30 and 32-42 as pending is presented in the Appendix.

4. Status of Amendments

Claims 1, 3-16, 18-30 and 32-42 were not amended after Final Rejection.

5. Summary of Claimed Subject Matter

A method for profiling computer program executions in a computer processing system having a processor and a memory hierarchy is claimed in claim 1. The method includes the following steps. A computer program is executed. (Specification, p. 19, line 7; p. 20, line 17). Profile counts are stored in a memory array. The profile counts are for a plurality of events associated with the execution of the computer program. The memory array is separate and distinct from the memory hierarchy so as to not perturb normal

operations of the memory hierarchy. (Specification, p. 14, lines 8-13; Figure 1, profile matrix 100; Figure 5, profile matrix 100). At least one of the plurality of events is selected for profiling. (Specification, p. 19, lines 11-14; Figure 3, step 304; Figure 4, step 404). The profile counts are updated for only the selected events. (Specification, p. 19, lines 20-21; p. 17, lines 15-22). Compilation and optimization of the computer program is assisted, based upon the selected profile counts stored in the memory array. (Specification, p. 22, line 22-p. 26, line 14; p. 28, line 19-p.29, line 8).

An apparatus for profiling computer program executions in a computer processing system having a processor and a memory hierarchy is claimed in claim 23. The apparatus includes the following. A memory array is provided that is adapted to store profile counts for events associated with execution of the computer program. The memory array is separate and distinct from the memory hierarchy so as to not perturb normal operations of the memory hierarchy. (Specification, p. 14, lines 8-13; Figure 1, profile matrix 100; Figure 5, profile matrix 100). A controller is provided that is adapted to select the events for profiling and to update the profile counts of the selected events stored in said memory array. (Specification, p. 19, lines 11-23; Figure 3, steps, 304, 310; Figure 4, step 404, 410). A scaling circuit is provided that is adapted to scale the profile counts to prevent profile information overflow. (Specification, p. 15, lines 14-24). The computer processing system assists compilation of the computer program, based upon the profile counts stored in the memory array. (Specification, p. 22, line 22-p. 26, line 14; p. 28, line 19-p.29, line 8).

A method for profiling computer program executions in a computer processing system having a processor and a memory hierarchy is claimed in claim 40. The method

includes the following steps. A computer program is executed. (Specification, p. 19, line 7; p. 20, line 17). Event-specific profile counts are stored in a memory array. Each profile count is associated with an event associated with the execution of a path of the computer program. The memory array is separate and distinct from the memory hierarchy so as to not perturb normal operations of the memory hierarchy. (Specification, p. 14, lines 8-13; Figure 1, profile matrix 100; Figure 5, profile matrix 100). At least one of the plurality of event-specific profile counts is selected is selected for profiling the path of the computer program. (Specification, p. 19, lines 11-14; Figure 3, step 304; Figure 4, step 404). If at least one of the selected event-specific profile counts has exceeded a predefined threshold, the portions of the computer program associated with the event-specific profile counts are optimized more aggressively than other portions of the computer program. (Specification, p. 22, lines 2-8; p. 22, line 22-p. 26, line 14; p. 28, line 19-p.29, line 8).

6. Grounds of Rejection to be Reviewed on Appeal

A. Claims 1, 4-8, 11-13, 16, 22, 38 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Krishnaswamy et al. (U.S. Patent No. 6.622,300) (hereinafter "Krishnaswamy").

B. Claims 3, 9-10, 23-30, 32-34, 37 and 39 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Krishnaswamy in view of "Dictionary of Computing" (hereinafter "Dictionary").

C. Claims 40 and 42 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Krishnaswamy in view of Chang et al. "Using Profile Information to Assist Classic Code Optimizations" (hereinafter "Chang").

7. Argument

A. Introduction

In rejecting claims under 35 U.S.C. § 103, the Examiner bears the burden of presenting a *prima facie* case of obviousness. *In re Rijckaert*, 9 F.3d 1531, 1532 (Fed. Cir. 1993). The burden of presenting a *prima facie* case of obviousness is only satisfied by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references. *In re Fine*, 837 F.2d 1071, 1074 (Fed. Cir. 1988). A *prima facie* case of obviousness is established when the teachings of the prior art itself would appear to have suggested the claimed subject matter to one of ordinary skill in the art. *In re Bell*, 991 F.2d 781, 782 (Fed. Cir. 1993). The suggestion to combine the references should come from the prior art, and the Examiner cannot use hindsight gleaned from the invention itself to pick and choose among related disclosures in the prior art to arrive at the claimed invention. *In re Fine*, 837 F.2d at 1075. If the Examiner fails to establish a *prima facie* case, the rejection is improper and must be overturned. *In re Rijckaert*, 9 F.3d at 1532 (citing *In re Fine*, 837 F.2d at 1074).

It is respectfully submitted that at the very least, the reference of Krishnaswamy is legally insufficient to sustain a *prima facie* case of obviousness against independent claim 1. Further, it is respectfully submitted that at the very least, the references of Krishnaswamy and Dictionary, taken individually or in any combination, are legally insufficient to sustain a *prima facie* case of obviousness against independent claim 23. Additionally, it is respectfully submitted that at the very least, the references of

Krishnaswamy and Chang, taken individually or in any combination, are legally insufficient to sustain a *prima facie* case of obviousness against independent claim 40.

For the reasons set forth below, Appellants respectfully request that the claim rejections under 35 U.S.C. § 103(a) be reversed.

B. Claims 1, 4-8, 11-13, 16, 22, 38 stand rejected as being unpatentable over Krishnaswamy.

(i). Krishnaswamy fails to teach or suggest "selecting at least one of the plurality of events for profiling," as claimed in claim 1.

Claim 1 claims, *inter alia*, "selecting at least one of the plurality of events for profiling." The Examiner admits on page 3 of the Final Office Action (Paper no. 112904) that "Krishnaswamy's background did not explicitly state selected events." The Examiner then cites col. 6, lines 21-30 of Krishnaswamy as demonstrating "selecting at least one of the purality of events for profiling. In the Advisory Action mailed on April 6, 2005, the Examiner specifically points to the word "programmable" in the phrase "programmable to count events like," as disclosed in col. lines 24-28 of Krishaswamy. The Examiner seems to assume that the statement "*programmable* to count events like" necessarily teaches or suggests "*selecting* at least one of the plurality of events for profiling." The term "programmable" does not necessarily imply a particular function. In particular, the term "programmable" does not imply that a selection of events is necessarily accomplished. For example, the counters for Krishnaswamy can be programmed to count events A, B and C. However, that the counters can be programmed to count events A, B and C does not necessarily imply that only A can be selected, or that only A and B can be selected, or any of a variety of combinations for that matter. Basically, the Examiner is contending that the

word "programmable" can be broadly interpreted to mean any function under the sun. Clearly, such a misunderstanding is without merit, facially unreasonable, and cannot be allowed.

Because Krishnaswamy neither teaches nor suggests each and every element of claim 1, it is respectfully asserted that no *prima facie* case of obviousness has been made out. Accordingly, the rejection of claims 1, 3-16, 18-22 and 38 should be reversed.

(ii). Krishnaswamy fails to teach or suggest "updating the profile counts for only the selected events" and "assisting compilation and optimization of the computer program, based upon the selected profile counts stored in the memory array," as claimed in claim 1.

Because Krishnaswamy does not teach or suggest "selecting at least one of the plurality of events for profiling," as showin in part (B)(i) above, it logically follows that Krishaswamy does not disclose "updating the profile counts *for only the selected events*" and "assisting compilation and optimization of the computer program, *based upon the selected profile counts* stored in the memory array."

Because Krishnaswamy neither teaches nor suggests each and every element of claim 1, it is respectfully asserted that no *prima facie* case of obviousness has been made out. Accordingly, the rejection of claims 1, 3-16, 18-22 and 38 should be reversed.

(iii). Krishnaswamy fails to teach or suggest "storing, in a memory array, profile counts for a plurality of events associated with the execution of the computer program, the memory array being separate and distinct from the memory hierarchy so as to not perturb normal operations of the memory hierarchy," as claimed in claim 1.

The Examiner incorrectly contends that counters of the PMU 90 in Krishnaswamy teach or suggest the "memory array," as claimed by Appellants. Such an assertion is

untenable and incorrect in the view of those skilled in the art. Although the Examiner is allowed a broad interpretation of the claim, the Examiner apparently forgets that the interpretation must be *reasonable*. No one skilled in the art would confuse the terms "memory" and "counters," as the Examiner contends.

A "memory" (and "memory array") as used in the general terminology of computer architecture is distinguishable from a "counter." Common examples of memory include SRAM and DRAM. On the other hand, a "counter" is built with latches or flip flops, which are "clocked elements." It should further be noted that while memory arrays are addressed, counters *cannot* be addressed, in contradiction to the description of "memory" in the patent application and also as claimed in claim 33.

Because Krishnaswamy neither teaches nor suggests each and every element of claim 1, it is respectfully asserted that no *prima facie* case of obviousness has been made out. Accordingly, the rejection of claims 1, 3-16, 18-22 and 38 should be reversed.

C. Claims 3, 9-10, 23-30, 32-34, 37 and 39 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Krishnaswamy in view of Dictionary.

(i). The combination of Krishnaswamy and Dictionary fails to teach or suggest "a controller adapted to select the events for profiling," as claimed in claim 23.

The Examiner applies the same rejection applied for "selecting at least one of the plurality of events for profiling," as claimed in claim 1, for "a controller adapted to select the events for profiling," as claimed in claim 23. Appellants submit that the arguments provided in section (B)(i) above apply similarly to this section. In particular, that the

counters in Krishnaswamy are "programmable" does not necessarily imply that a selection of the events is made.

Because the combination Krishnaswamy and Dictionary neither teaches nor suggests each and every element of claim 23, it is respectfully asserted that no *prima facie* case of obviousness has been made out. Accordingly, the rejection of claims 23-30 and 32-39 should be reversed.

(ii). The combination of Krishnaswamy and Dictionary fails to teach or suggest "a controller adapted...to update the profile counts of the selected events stored in said memory array" and "wherein the computer processing system assists compilation of the computer program, based upon the profile counts stored in the memory array," as claimed in claim 23.

Because the combination of Krishnaswamy and Dictionary does not teach or suggest "selecting at least one of the plurality of events for profiling," as showin in part (C)(i) above, it logically follows that the combination of Krishaswamy and Dictionary does not disclose "a controller adapted...to update the profile counts of the selected events stored in said memory array" and "wherein the computer processing system assists compilation of the computer program, based upon the profile counts stored in the memory array."

Because the combination of Krishnaswamy and Dictionary neither teaches nor suggests each and every element of claim 23, it is respectfully asserted that no *prima facie* case of obviousness has been made out. Accordingly, the rejection of claims 23-30 and 32-39 should be reversed.

(iii). The combination of Krishnaswamy and Dictionary fails to teach or suggest "a scaling circuit adapted to scale the profile counts to prevent profile information overflow," as claimed in claim 23 and essentially as claimed in claim 10.

The Examiner cites Dictionary as disclosing "a scaling circuit adapted to scale the profile counts to prevent profile information overflow," as claimed in claim 23 and as essentially claimed in claim 10. A generic definition of scaling does not provide for a scaling circuit or scaling functionality in the claims. Further, there is no clearer example of hindsight reconstruction than the one provided by the Examiner here. Nothing in Krishnaswamy or Dictionary indicate any motivation, teaching or suggestion for such a combination. The Examiner in hindsight merely found a definition in Dictionary to satisfy his rejection. It is impermissible to use the claimed invention as an instruction manual or "template" in attempting to piece together isolated disclosures and teachings of the prior art so that the claimed invention is rendered obvious. *In re Fritch*, 972 F.2d 1260, 1266, 23 USPQ2d 1780, 1783 (Fed. Cir. 1992). There is simply no other way to explain the disparate citation to Dictionary other than improper hindsight reasoning.

The Examiner states, without support, that it "would have been obvious [to combine Krishnaswamy with Dictionary] because one of ordinary skill in the art would be motivated to adjust the stored value to the hardware/equipment (register size limitations) (Computing, page 432)." It is entirely unclear to Appellants how the citation to page 432 of Dictionary supports the Examiner's statement. It is further unclear where such reasoning originated, other than from the Examiner's own imagination. The statement by the Examiner is clearly conclusory and without merit.

Because the combination Krishnaswamy and Dictionary neither teaches nor suggests each and every element of claims 23 and 10, it is respectfully asserted that no

prima facie case of obviousness has been made out. Accordingly, the rejection of claims 3, 9-10, 23-30 and 32-39 should be reversed.

(iv). The combination of Krishnaswamy and Dictionary fails to teach or suggest "a memory array adapted to store profile counts for events associated with execution of the computer program, said memory array being separate and distinct from the memory hierarchy so as to not perturb normal operations of the memory hierarchy," as claimed in claim 23.

The Examiner incorrectly contends that counters of the PMU 90 in Krishnaswamy teach or suggest the "memory array," as claimed by Appellants. Such an assertion is untenable and incorrect in the view of those skilled in the art. Although the Examiner is allowed a broad interpretation of the claim, the Examiner apparently forgets that the interpretation must be *reasonable*. No one skilled in the art would confuse the terms "memory" and "counters," as the Examiner contends.

A "memory" (and "memory array") as used in the general terminology of computer architecture is distinguishable from a "counter." Common examples of memory include SRAM and DRAM. On the other hand, a "counter" is built with latches or flip flops, which are "clocked elements." It should further be noted that while memory arrays are addressed, counters *cannot* be addressed, in contradiction to the description of "memory" in the patent application and also as claimed in claim 33.

Because the combination of Krishnaswamy and Dictionary neither teaches nor suggests each and every element of claim 23, it is respectfully asserted that no *prima facie* case of obviousness has been made out. Accordingly, the rejection of claims 23-30 and 32-39 should be reversed.

D. Claims 40 and 42 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Krishnaswamy in view of Chang.

(i). The combination of Krishnaswamy and Chang fails to teach or suggest "selecting at least one of the plurality of event-specific profile counts for profiling the path of the computer program," as claimed in claim 40.

The Examiner applies the same rejection applied for "selecting at least one of the plurality of events for profiling," as claimed in claim 1, for "selecting at least one of the plurality of event-specific profile counts for profiling the path of the computer program" as claimed in claim 40. Appellants submit that the arguments provided in section (B)(i) above apply similarly to this section. In particular, that the counters in Krishnaswamy are "programmable" does not necessarily imply that a selection of the events is made.

Because the combination of Krishnaswamy and Chang neither teaches nor suggests each and every element of claim 40, it is respectfully asserted that no *prima facie* case of obviousness has been made out. Accordingly, the rejection of claims 40-42 should be reversed.

(ii). The combination of Krishnaswamy and Chang fails to teach or suggest "if at least one of the selected event-specific profile counts has exceeded a predefined threshold, optimizing the portions of the computer program associated with the event-specific profile counts more aggressively than other portions of the computer program," as claimed in claim 40.

Because the combination of Krishnaswamy and Chang does not teach or suggest "selecting at least one of the plurality of event-specific profile counts for profiling the path of the computer program," as showin in part (D)(i) above, it logically follows that the combination of Krishaswamy and Chang does not disclose "if at least one of *the selected*

event-specific profile counts has exceeded a predefined threshold, optimizing the portions of the computer program associated with the event-specific profile counts more aggressively than other portions of the computer program."

Because the combination of Krishnaswamy and Chang neither teaches nor suggests each and every element of claim 40, it is respectfully asserted that no *prima facie* case of obviousness has been made out. Accordingly, the rejection of claims 40-42 should be reversed.

(iii). No proper suggestion, teaching or motivation is provided for the combination of Krishnaswamy and Chang.

In combining Krishnaswamy and Chang, the Examiner states, without support, that the combination "would have been obvious because one of ordinary skill in the art would be motivated to optimize frequently executed program paths primarily since they are executed more (page 1301, Introduction and pages 1306-1308)." The citation to Chang does not provide any motivation or suggest to combine the optimizations of Chang with the PMU counters of Krishnaswamy. Again, as with section (C)(iii) above, the only plausible support for combining Krishnaswamy and Chang in such a manner is improper hindsight reasoning using the Appellants' disclosure.

Because no proper motivation, teaching or suggstion is provided for the combination of Krishnaswamy and Chang, it is respectfully asserted that no *prima facie* case of obviousness has been made out. Accordingly, the rejection of claims 40-42 should be reversed.

(iv). **The combination of Krishnaswamy and Chang fails to teach or suggest "storing, in a memory array, a plurality of event-specific profile counts, each associated with an event associated with the execution of a path of the computer program, the memory array being separate and distinct from the memory hierarchy so as to not perturb normal operations of the memory hierarchy," as claimed in claim 40.**

The Examiner incorrectly contends that counters of the PMU 90 in Krishnaswamy teach or suggest the "memory array," as claimed by Appellants. Such an assertion is untenable and incorrect in the view of those skilled in the art. Although the Examiner is allowed a broad interpretation of the claim, the Examiner apparently forgets that the interpretation must be *reasonable*. No one skilled in the art would confuse the terms "memory" and "counters," as the Examiner contends.

A "memory" (and "memory array") as used in the general terminology of computer architecture is distinguishable from a "counter." Common examples of memory include SRAM and DRAM. On the other hand, a "counter" is built with latches or flip flops, which are "clocked elements." It should further be noted that while memory arrays are addressed, counters *cannot* be addressed, in contradiction to the description of "memory" in the patent application and also as claimed in claim 33.

Because the combination of Krishnaswamy and Chang neither teaches nor suggests each and every element of claim 40, it is respectfully asserted that no *prima facie* case of obviousness has been made out. Accordingly, the rejection of claims 40-42 should be reversed.

F. CONCLUSION

The claimed invention is not disclosed or suggested by the teachings of the applied prior art references, either alone or in combination. Moreover, the Examiner has failed to establish a *prima facie* case of obviousness of the presently claimed method under 35 U.S.C. § 103(a) over the combination of Krishnaswamy with various other references for at least the reasons noted above. Accordingly, it is respectfully requested that the Board reverse the rejection of claims 1, 3-16, 18-30 and 32-42 under 35 U.S.C. § 103(a).

Respectfully submitted,

By: _____

Koon Hon Wong
Reg. No. 48,459
Attorney for Appellants

F. Chau & Associates, LLC
130 Woodbury Road
Woodbury, NY 11797
TEL: (516) 692-8888
FAX: (516) 692-8889
Attorneys for Appellants

Claims Appendix

1. A method for profiling computer program executions in a computer processing system having a processor and a memory hierarchy, comprising the steps of:

executing a computer program;

storing, in a memory array, profile counts for a plurality of events associated with the execution of the computer program, the memory array being separate and distinct from the memory hierarchy so as to not perturb normal operations of the memory hierarchy;

selecting at least one of the plurality of events for profiling;

updating the profile counts for only the selected events;

assisting compilation and optimization of the computer program, based upon the selected profile counts stored in the memory array.

2. (Cancelled).

3. The method according to claim 1, wherein said storing and updating steps are performed asynchronously to prevent a decrease of an execution speed of the computer program.

4. The method according to claim 1, wherein said updating step is triggered by execution of the events.

5. The method according to claim 1, wherein said updating step is triggered by execution of instructions embedded into an instruction stream of the computer program.

6. he method according to claim 1, further comprising the step of detecting whether a profile count has exceeded an adjustable predefined threshold.

7. The method according to claim 1, further comprising the step of indicating when a profile count has exceeded an adjustable predefined threshold.

8. The method according to claim 7, wherein said indicating step comprises the step of raising an exception.

9. The method according to claim 1, further comprising the steps of:
accumulating profile updates; and
dividing the accumulated profile updates by a threshold fraction.

10. The method according to claim 1, further comprising the step of scaling the profile counts to prevent profile information overflow.

11. The method according to claim 1, further comprising the step of identifying profile information corresponding to the profile counts using a profiling event identifier.

12. The method according to claim 11, further comprising the step of addressing the memory array, using the profiling event identifier.

13. The method according to claim 1, further comprising the steps of:
generating the profile counts using profile counters associated with the events; and
maintaining the profile counters in a set-associate manner.

14. The method according to claim 13, further comprising the step of selecting a profile counter to be evicted from the memory array based upon a predefined replacement, when a number of profiling events assigned to an associative class of events is exceeded.

15. The method according to claim 14, wherein the replacement strategy is based upon one of least-recently-used and first-in-first-out.

16. The method according to claim 1, further comprising the step of supporting read operations from the memory array in an off-line optimization of the program.

17. (Cancelled).

18. The method according to claim 1, wherein said assisting step is performed during at least one of dynamic binary translation and dynamic optimization of the computer program.

19. The method according to claim 18, wherein the dynamic binary translation and dynamic optimization of the computer program results in translated and optimized code, respectively, the translated and optimized code comprising instructions groups which pass control therebetween.

20. The method according to claim 19, further comprising the step of identifying frequently executed paths of the computer program, by instrumenting exits from the instruction groups with a profiling instruction that indicates a unique group exit identifier.

21. The method according to claim 19, further comprising the step of extending the instruction groups along a frequently executed path.

22. The method according to claim 1, wherein the memory hierarchy includes data and instruction caches, and the memory array is separate and distinct from the memory hierarchy so as to not perturb normal operations of the data and instruction caches.

23. An apparatus for profiling computer program executions in a computer processing system having a processor and a memory hierarchy, the apparatus comprising:

a memory array adapted to store profile counts for events associated with execution of the computer program, said memory array being separate and distinct from the memory hierarchy so as to not perturb normal operations of the memory hierarchy;

a controller adapted to select the events for profiling and to update the profile counts of the selected events stored in said memory array; and

a scaling circuit adapted to scale the profile counts to prevent profile information overflow;

wherein the computer processing system assists compilation of the computer program, based upon the profile counts stored in the memory array.

24. The apparatus according to claim 23, wherein said memory array and said controller are adapted to asynchronously store and update the profile counts, respectively, to prevent a decrease of an execution speed of the computer program.

25. The apparatus according to claim 23, wherein said controller is adapted to update the profile counts as the events are executed.

26. The apparatus according to claim 23, wherein said controller is adapted to update the profile counts based upon instructions embedded into an instruction stream of the computer program.

27. The apparatus according to claim 23, further comprising a comparator circuit adapted to detect whether a profile count has exceeded an adjustable predefined threshold.

28. The apparatus according to claim 23, further comprising an indicating circuit for indicating when a profile count has exceeded an adjustable predefined threshold.

29. The apparatus according to claim 28, wherein said indicating circuit is adapted to raise an exception when the profile count has exceeded the adjustable predefined threshold.

30. The apparatus according to claim 23, further comprising:

an accumulation circuit adapted to accumulate the updated profile counts; and

a dividing circuit adapted to divide an accumulated value of the updated accumulated profile counts by a threshold fraction.

31. (Cancelled).

32. The apparatus according to claim 23, wherein profile information corresponding to the profile counts is identified using a profiling event identifier.

33. The apparatus according to claim 32, wherein the memory array is addressed using the profiling event identifier.

34. The apparatus according to claim 23, further comprising profile counters for generating the profile counts, said profile counters being associated with an event in a set-associate manner.

35. The apparatus according to claim 34, further comprising a replacement circuit adapted to select a profile counter to be evicted from the memory array based on a predefined replacement strategy, when a number of profiling events assigned to an associative class is exceeded.

36. The apparatus according to claim 35, wherein the predefined replacement strategy is based upon one of least-recently-used and first-in-first-out.

37. The apparatus according to claim 23, wherein the memory hierarchy includes data and instruction caches, and said memory array is separate and distinct from the memory hierarchy so as to not perturb normal operations of the data and instruction caches.

38. The method according to claim 1, wherein said method is implemented by a program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform said method steps.

39. The apparatus according to claim 23, further comprising wherein the computer processing system assists optimization of the computer program, based upon the profile counts stored in the memory array.

40. A method for profiling computer program executions in a computer processing system having a processor and a memory hierarchy, comprising the steps of:

executing a computer program;

storing, in a memory array, a plurality of event-specific profile counts, each associated with an event associated with the execution of a path of the computer program, the memory array being separate and distinct from the memory hierarchy so as to not perturb normal operations of the memory hierarchy;

selecting at least one of the plurality of event-specific profile counts for profiling the path of the computer program; and

if at least one of the selected event-specific profile counts has exceeded a predefined threshold, optimizing the portions of the computer program associated with the event-specific profile counts more aggressively than other portions of the computer program.

41. The method according to claim 40, further comprising the step of optimizing of the portions of the computer program during at least one of static and dynamic compilation.

42. The method according to claim 40, wherein the memory array is arranged as a two-way set associative array.

Evidence Appendix

None

Related Procedings Appendix

None